# METHOD AND APPARATUS TO SUPPORT AN EXPANDED REGISTER SET

## FIELD OF THE INVENTION

Embodiments of the present invention relate to microprocessors. More particularly, embodiments of the present invention relate to an expanded set of registers in a microprocessor.

## BACKGROUND OF THE INVENTION

A known microprocessor architecture is the IA-32 architecture of Intel Corp. of Santa Clara, California. The IA-32 architecture includes 32-bit general integer registers. The general integer registers can be used as operands for calculations and for addressing. For example, a register can store a value that is part of an operation (e.g., an arithmetic operation, a logical operation, etc.). In another example, a register can store information relating to a memory address that stores a value that is part of an operation. The IA-32 architecture includes a small number of logical general integer registers, i.e., eight logical general integer registers. In general, fewer logical registers can disadvantageously ~~effect~~ affect system performance because software compilers can be limited in terms of optimizations that can be implemented. For example, fewer

registers can require increased accesses to memory and/or stack, which can decrease system performance. In view of the foregoing, it can be appreciated that a substantial need exists for methods and apparatuses which can advantageously support an expanded logical register set.

## BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 shows an example of an instruction form of an IA-32 instruction.

FIG. 2 shows an example of an instruction form of an IA-32 instruction.

FIG. 3 shows an example of an instruction form in accordance with an embodiment of the present invention.

FIG. 4 shows an example of an instruction form in accordance with an embodiment of the present invention.

FIG. 5 shows an example of an instruction form in accordance with an embodiment of the present invention.

FIG. 6 shows data in accordance with embodiments of the present invention.

FIG. 7 shows an apparatus in accordance with an embodiment of the present invention.

## DETAILED DESCRIPTION

According to an embodiment of the present invention, a microprocessor includes an expanded logical register set that can be accessed by instructions including legacy opcodes and remapped addressing mode information. The known IA-32 instruction set is limited to accessing eight logical general integer registers. An IA-32 instruction can specify which of the eight logical general integer registers are to be accessed via 3-bit register identifier fields of the addressing mode information of the

instruction. Each 3-bit register identifier can specify any of the eight logical general integer registers. An expanded logical register set (e.g., sixteen logical registers, thirty-two logical registers, sixty-four logical registers, etc.) can be accessed by remapping the addressing mode information to include at least four-bit register identifiers without defining new opcodes or defining additional instruction prefixes.

FIG. 1 shows an example of a general instruction format of an IA-32 instruction. IA-32 instruction encodings can be subsets of the general instruction format shown in FIG. 1. An IA-32 instruction can include up to four optional prefix(es) 110, an opcode 120, a ModR/M byte 130, a SIB (scale index base) byte 140, an optional address displacement 150, and an optional immediate operand 160.

The optional prefix(es) 110 can include a byte, for example, that specifies that the instruction utilizes Streaming SIMD (single instruction multiple data) Extensions that can be executed, for example, by a Pentium® III processor manufactured by Intel Corp. The opcode 120 can include a primary opcode of one or two bytes.

ModR/M byte 130 and the SIB byte 140 can include addressing mode information. The ModR/M byte 130 can include a mod field 136, a reg/opcode field 134, and an r/m field 132. The mod field 136 can combine with the r/m field 132 to specify thirty-two values specifying eight registers and twenty-four addressing modes. The r/m field 132 can specify a register as an operand or can be combined with the mod field 136 to encode an addressing mode. The reg/opcode field 134 can include a register identifier to specify a register number or three additional bits of opcode information, and the reg/opcode field's purpose (e.g., a reg field, an opcode field, etc) can be specified within opcode 120. Encodings of the ModR/M byte 130 can require a second addressing mode information byte -- the SIB byte 140 -- to specify the addressing mode of an instruction. For example, the SIB byte 140 can be required in

certain 32-bit addressing forms of the IA-32 architecture, such as the base-plus-index addressing form, the scale-plus-index addressing form, etc. The SIB byte can include a scale field 146 to specify a scale factor, an index field 144 including a register identifier to specify a register number of an index register, and a base field 142 including a register identifier to specify a register number of a base register.

An IA-32 instruction can include a displacement 150 for certain addressing forms. When an addressing form includes a displacement 150, the displacement 150 can include a plurality of bytes, e.g., one byte, two bytes, four bytes, etc. An IA-32 can also include an immediate operand 160, which can follow any displacement 150. The immediate operand 160 can include one byte, two bytes, four bytes, etc. The optional immediate operand 160, when present, can include n bits (e.g., eight bits, sixteen bits, thirty-two bits, etc.).

FIG. 2 shows an example of a particular instruction format of an IA-32 instruction. Instruction 200 includes ~~an~~ a ModR/M byte ~~240~~ 230 with a mod field 236 having a value of 01B ("01B" encompasses a binary value of 01) and an r/m field 232 having a value of 100B. A SIB byte ~~140~~ 240 of instruction 200 includes an index field 244 having a value of 100B. The reg/opcode field 134 includes three bits (bits 3 through 5 of ModR/M byte 230) that can specify one of eight logical general integer registers, and base field 142 includes three bits (bits 0 through 2 of SIB byte 240) that can specify one of eight logical general integer registers. An instruction having the format of instruction 200 -- where the ~~ModR/M field~~ mod field 236 is 01B, the R/M field 232 is 100B, the index field is 100B -- has addressing mode information that is currently unsupported in the IA-32 architecture regardless of the value of the scale field 246.